Colored Noise Functions for Stochastic Analysis

Miro Stoyanov, Max Gunzburger, John Burkardt
Department of Scientific Computing,
Florida State University, Tallahassee, Florida

February 16, 2024

1 Introduction

The modern discipline of uncertainty quantification seeks to understand and estimate the effect of approximated and unobserved factors on a system. A physical system is described by a deterministic mathematical equation which might have the form

$$\mathcal{L}(u(x,t)) = f(x,t)$$

and which intends to model the behavior of the state variable u which is governed by the operator $\mathcal{L}$, the forcing term f and additional problem data such as boundary or initial conditions. If the operator $\mathcal{L}$ blah blah **some reference will be good here**, then for a given range of problem data it is theoretically possible to determine the state function u.

In practice, however, there is a number of reasons to question whether such deterministic models give sufficient justification to trust the answer. It may be the case that the data of the physical system, such as the function f, is only known at selected values, and is interpolated elsewhere. Physical parameters such as temperature may only be known to a few digits of accuracy. The physics that is embodied in the operator $\mathcal{L}$ may also be only approximately understood or described. Finally, the process of determining the solution of the mathematical equation will generally involve finitely accurate arithmetic and iterations that only produce approximate solutions. Thus, there may be a long and involved process between the observation of physical data f and the determination that the corresponding state has been computed to be u.

Cases are known in which small errors or changes in the problem specification can result in disastrous changes in the computed solution. Since the equations being solved may describe aircraft wings, building supports, or medical devices, it is vital to demonstrate that the computational results are a reliable estimate of the physical behavior to be expected.

The standard technique for studying the influence of uncertainty and error on a mathematical model involves the use of what is termed *noise*. We may represent this formally as a function $\eta(x,t;\omega)$; here, the ω indicates the dependence of the noise on some unobservable underlying process. For a particular

ω, we get a *realization* of the noise function, that is, a particular example of the many kinds of possible noise. In the simplest sort of study, we might imagine that this noise function represents uncertainty or error in the right hand side $f(x,t)$, so that we now consider the problem of solving a system of the form

$$\mathcal{L}(u(x,t;\omega)) = f(x,t) + \eta(x,t;\omega)$$

The purpose of the ω argument is to indicate the underlying randomness of the noise function. In order to study the influence of the noise on the resulting perturbed state function $u(x,t:\omega)$, we really need to consider all the noise functions possible over the range of ω. The standard way of doing this requires that we propose a probability density function to be associated with ω. If such a PDF can be supplied, it is possible to produce statistical moments that indicate the effect of the noise. In particular, we may compute the expected value of the solution under the influence of noise:

$$E(u(x,t;\omega)) = \int_{\Omega} u(x,t;\omega) d\Omega$$

The variance of the solution can be computed in a similar fashion. If we are able to compute just these two quantities, then in contrast to the classical solution of the deterministic problem, we now have a solution value that has averaged the effects of all possible noise perturbations, as well as a measure of the typical variation from that base solution that we can expect for the solution associated with any particular single realization of the noise function.

2 White Noise

In signal analysis, noise is defined as a signal that does not carry information. When thinking of the original physical system, noise is the unmeasurable, unobservable, uncontrollable variations from the physical data we can observe. In the mathematical model, we include a corresponding function $\eta(x,t;\omega)$ which is intended to suggest that some elements of the model are not known precisely, or must be allowed to vary.

Since this is a mathematical model, we are free to choose any representation for these perturbations or unobservable quantities that we find appropriately imitates the behaviors found in the physical model. The most common choice is to choose a *white noise* function. The value of a white noise function at any point is taken to be random, and to have expected value 0. Moreover, the values of the function at any two distinct points are assumed to be independent and uncorrelated. Such a noise function may seem the natural representation of totally random variation.

The value that the noise function takes at any point and time is drawn from some prescribed probability density function (PDF). A common choice is to use a Gaussian PDF. Choosing the Gaussian PDF allows a nonzero probability that the noise may, locally and momentarily, assume a value of arbitrarily large modulus. Other choices for the PDF may be made, including a simple uniform density in [-1,+1], for instance, allowing control over the range and variance of the values assumed locally by the noise.

Computationally, it is necessary to discretize the mathematical model. Such quantities as the right hand side function f and the state variable u might now be represented by mesh functions defined only at a grid of points (x_i, t_j), and hence values of the noise function will also only be required at discrete points in space and time. For convenience, we will assume a grid sizes of Δx spatially and Δt temporally. In order to allow for consistent behavior as we vary the grid sizes, we will expect the noise function at a point (x_i, t_j) to have the form

$$\eta_{i,j} = \eta(x_i, t_j) = \frac{\sigma^2}{\sqrt{\Delta t}\sqrt{\Delta x}} y_{i,j}$$

where σ controls the variance, and the $y_{i,j}$ values might be independent Gaussian samples with zero mean and unit variance. From this formula, it is easy to see how to computationally represent a discretized version of white noise.

White noise may seem the natural way to model uncertainty, and it is quite easy to compute samples of discretized white noise computationally. However, there are some features of white noise that can be regarded as drawbacks or flaws. Perhaps the first problem to note occurs when we consider the noise function associated with the undiscretized problem. If we fix our attention on a single spatial location, the white noise at that point becomes purely a function of time. The construction of the white noise function implies that the power spectrum is a constant function of frequency, or, in other words, there is an equal amount of energy associated with every octave. Should this be strictly true, any nonzero white noise signal must have infinite energy. This, in turn, means the sequence of white noise functions associated with an increasingly refined temporal grid will have energy that grows unboundedly. Whether or not this is a computational problem, it does indicate that there is necessarily a weakness in the correspondence between the mathematical noise and whatever physical phenomenon it is intended to model.

3 Colored Noise

Alternatives to white noise have been proposed; in particular, a family of *power law* noise functions have been proposed; the name is meant to indicate the form of the power density function. Many natural phenomena that can be modeled by a time series exhibit behavior similar to what is termed $1/f$ *noise* or *pink noise*. This is a noise signal whose power density function, over some significant range, is proportional to the inverse of the frequency.

If we assume that the power density spectrum of a noise signal can be modeled as $1/f^\alpha$, then white noise has $\alpha = 0$, pink noise has $\alpha = 1$ and it turns out that Brownian motion corresponds to $\alpha = 2$. Pink noise thus lies "halfway" between uncorrelated white noise and highly correlated Brownian motion. Moreover, we are free to consider other members of this power law noise family with

$\alpha = \frac{2}{3}$ or $\alpha = 1.5$, for instance, and the noise functions corresponding to the range $0 \leq \alpha \leq 2$ is termed the family of *colored noise* functions.

In stochastic analysis, these alternative noise functions have the same role as the standard white noise; however, for a given problem they might prove more appropriate as an approximation to a disturbance whose power spectrum is known, their spectral properties might prove more tractable or useful, and their integrability may be an advantage. In any case, it is worthwhile to realize that white noise is not the only noise function that can be chosen, and that the choice of noise function can have an influence on the results of a stochastic analysis.

In particular, we find it of interest to compare some of these power law noise functions. The simplest thing to consider is the behavior of the variance of the mean of realizations of a noise function associated with a particular α. Somewhat more unpredictable is the influence on the expected value and variance of state variables computed from a stochastic equation governed by noise of the particular type. It may be enough, in this discussion, to note that there *are* differences.

We turn now to the question of how, for a chosen value of α, it is possible to generate a realization of a colored noise function.

4 Miro Tells The Story: Implementation of α noise

We consider the algorithm of Kasdin [2] for generating discrete white noise. The idea is to generate a white noise vector, by sampling from an underlying zero-mean Gaussian distribution, then to take a convolution of that vector with a "weight" vector. Since discrete convolution has complexity $O(N^2)$, in the implementation we replace the convolution with Fast Fourier Transform that has complexity of $O(N \log(N))$. The original code from Kasdin uses "Numerical Recipes" real Fourier Transform procedure, however, implement the algorithm using MATLAB inbuild complex `fft()` function and thus we have to use same scaling to match the output of the original Kasdin's algorithm.

Algorithm: Coloured Noise Generator

Given N, the number of discrete points, σ, the standard deviation of the zero-mean Gaussian distribution from which we sample and α, the algorithm generates a noise vector $\boldsymbol{\eta} \in \mathbb{R}^N$ with frequency distribution $\frac{1}{f^\alpha}$

- First we generate a vector of weights $< h_i > \in \mathbb{R}^{2N}$

$$h_i = \begin{cases} 1.0 & i = 1, \\ h_{i-1} \frac{0.5\alpha + (i-2)}{i-1} & 2 \leq i \leq N, \\ 0.0 & N < i. \end{cases} \quad (1)$$

- Second generate a "white" noise vector $< \omega_i > \in \mathbb{R}^{2N}$, where each w_i is independently sampled from a Gaussian distribution with zero mean and standard deviation σ.
- Compute $< \hat{h}_i >, < \hat{\omega}_i > \in \mathbb{C}^{2N}$ as the Fast Fourier Transforms of $< h_i >$ and $< \omega_i >$ respectively.
- Let $< \hat{f}_i > \in \mathbb{C}^{2N}$ be the index wise product of $\hat{f}_i = \hat{h}_i \hat{\omega}_i$.
- Scale $\hat{f}_1 = \frac{\hat{f}_1}{2}$, $\hat{f}_{N+1} = \frac{\hat{f}_{N+1}}{2}$ and set $\hat{f}_i = 0.0$ for $i > N + 1$.
- Let $< f_i > \in \mathbb{C}^{2N}$ be the inverse Fourier Transform of $< \hat{f}_i >$, then the computed solution is

$$\eta_i = 2\Re(f_i), \text{ for } 1 \leq i \leq N. \tag{2}$$

We present the MATLAB code that generates $\frac{1}{f^\alpha}$ noise.

```
function [ eta ] = ff_alpha ( n, sigma, alpha )
  hfa = zeros ( 2 * n, 1 );
  hfa(1) = 1.0;
  for i = 2 : n
    hfa(i) = hfa(i-1) * ( 0.5 * alpha + ( i - 2 ) ) / ( i - 1 );
  end
  hfa(n+1:2*n) = 0.0;

  wfa = [ sigma * randn( n, 1 ); zeros( n, 1 ); ];

  [ fh ] = fft( hfa );
  [ fw ] = fft( wfa );

  fh = fh( 1:n + 1 );
  fw = fw( 1:n + 1 );

  fw = fh .* fw;

  fw(1)        = fw(1) / 2;
  fw(end)      = fw(end) / 2;
  fw = [ fw; zeros(n-1,1); ];

  eta = ifft( fw );
  eta = 2*real( eta(1:n) );
end
```

5 Numerical Results

The first question that we can ask is what is the efficiency of the algorithm. One of the main attractions of white noise is the simplicity of its implementation and

Table 1: Pure Noise

Standard Deviation of the mean of pure noise $\eta_i(N,\alpha)$. $\sigma = 1$							
$\alpha : N$	10	250	500	1000	10000	20000	50000
0.0	1.05590	0.99566	1.00440	0.98595	0.99790	0.99756	1.00580
0.5	0.93257	0.89965	0.90187	0.89221	0.90449	0.90211	0.89132
1.0	0.81766	0.79623	0.80413	0.79190	0.80806	0.79518	0.79209
1.5	0.70664	0.69500	0.68337	0.69347	0.69348	0.69266	0.68989
2.0	0.59378	0.57871	0.57827	0.58321	0.58059	0.58184	0.58419

the very fast way to compute a large number of white noise vectors. We compare the time it takes to compute $10,000$ pink-noise vectors vs the same number of white noise vectors. On MATLAB 2009b with AMD 3.4Ghz CPU and Linux operating system the colored-noise algorithm is approximately 13 times slower than the simple white noise generator. Since in practical problems generating noise is only a small part ofthe computation, the cost of 13 times will often be absorbed into more time consuming parts. Nevertheless, one should still ask if the price is worthy. So we try to investigate the statistical properties of $\frac{1}{f^\alpha}$ noise for different values of α.

For the first study, we fix the standard deviation of the underlying Gaussian distribution to be $\sigma = 1$. Then we consider the domain $\Omega = [0,1]$ and we select N uniformly distributed points, and for each α, we generate a number of noise vectors $\boldsymbol{\eta}, \omega_i(N,\alpha)$. Since each $\boldsymbol{\eta}, \omega_i(N,\alpha)$ is an approximation to a continuous $\boldsymbol{\eta}, \omega(x,\alpha)$, we have to account for the mesh density by scaling the vectors by $N^{\frac{1-\alpha}{2}}$. For white noise (i.e. $\alpha = 0$) we scale by $N^{\frac{1}{2}}$, which leads to a standard way of discretizing white noise (**add reference here**), and for Brownian motion (i.e. $\alpha = 2$) the scaling factor is $N^{-\frac{1}{2}}$, which is known (**add more reference here**). For each computed noise vector, the expected value of the mean is zero, however, we wish to study the variation of the mean. For each value of N and α, we generate $10,000$ sample noise vectors and we compute the standard deviation of the mean. We present the results on Table 1. It is clear that even though our sampling is based upon the same underlying Gaussian distribution, the statistical properties of the noise for different values of α are quite different. As α increases the noise has considerably lower variance.

Next we study the 1-D Laplace equation

$$u_{xx}(x) = x + \eta(\alpha,\omega), \qquad u(0) = 0, \qquad u(1) = \frac{1}{6}, \tag{3}$$

where $\eta(\alpha)$ is $\frac{1}{f^\alpha}$ noise. The noise free solution is $u(x) = \frac{1}{6}x^3$. We use finite difference approximation for the Laplacian operator on an uniform mesh with $N+1$ points and we use Algorithm (reference the algorithm) to generate noise and perturb the right hand side of the equation. For each N and α we solve $10,000$ equations and we gather statistics about the standard deviation of the

Table 2: Additive Noise

Standard Deviation of the mean of the solution $\mathbf{u}(N, \alpha)$. $\sigma = 1$							
$\alpha : N$	10	250	500	1000	10000	20000	50000
0.0	0.100660	0.092005	0.092463	0.092402	0.091159	0.090339	0.091812
0.5	0.089742	0.081129	0.082130	0.080622	0.081340	0.081260	0.081346
1.0	0.078451	0.072538	0.072242	0.072441	0.071642	0.072044	0.071774
1.5	0.068110	0.061172	0.061011	0.061572	0.062032	0.061570	0.061073
2.0	0.056753	0.051480	0.050756	0.050733	0.050878	0.051013	0.050569

Table 3: Computed Coefficients

Standard Deviation of the absolute value of individual Fourier coefficients $\mathbf{u}(N, \alpha)$. $\sigma = 1$						
$\alpha : k$	1	2	3	7	15	25
0.0	0.060275	0.0193360	0.00478520	$5.0207e-04$	$9.2034e-05$	$3.1170e-05$
0.5	0.054468	0.0155670	0.00302830	$2.1487e-04$	$3.0748e-05$	$9.0569e-06$
1.0	0.049850	0.0124460	0.00194620	$1.0256e-04$	$1.0927e-05$	$2.6943e-06$
1.5	0.043352	0.0096661	0.00136300	$5.7020e-05$	$4.8606e-06$	$1.0156e-06$
2.0	0.037937	0.0070716	0.00092786	$3.5221e-05$	$2.7887e-06$	$5.4937e-07$

mean of the solution. The results are given on Table 2. From the results, noise for different values of α appears to act as scaled "white" noise. It is natural to ask, if we can match the behaviour of the solution for different α, by simply "tweaking" the standard deviation σ. Indeed, we can make the variance of the mean identical, however, the difference between $\eta(0)$ and $\eta(\alpha > 0)$ is not only in the total variance, but also in the frequency distribution of the noise. To demonstrate that, we take the case of $N = 50000$ and we compute the Fourier Transforms of the solution vectors associated with different values of α. Then we consider the variance of the individual Fourier coefficients $\hat{u}_k$. Due to the eigenstructure of the Laplace operator, the lowest three frequencies contain the bulk of the energy of the system and the effect of α on $\hat{u}_1$, $\hat{u}_2$ and $\hat{u}_3$ seems to be that of near uniform scaling. However, when we look at the higher frequencies, we see that the drop-rate as k increases is much higher for larger values of α, which means that the solution is overall smoother. We conclude that for different values of α, the effect of the noise onto the computed solution extends beyond the simple scaling observed in Table 1.

Next we add the noise to the coefficient of the Laplace equation:

$$(1 + |\eta(x, \alpha)|)\, u_{xx}(x) = x, \qquad u(0) = 0, \qquad u(1) = \frac{1}{6}. \tag{4}$$

Note that we take the absolute value of $\eta(\alpha)$ to prevent the left hand side from

Table 4: Coefficient Noise Mean (absolute value)

Mean of the solution $\mathbf{u}(N,\alpha)$. $\sigma = 1$							
$\alpha : N$	10	250	500	1000	10000	20000	50000
0.0	0.013989	0.0057368	0.0049075	0.003852	0.0013853	0.0012235	0.00074086
0.5	0.018178	0.0135610	0.0120650	0.010761	0.0078303	0.0068163	0.00608480
1.0	0.020341	0.0207380	0.0205360	0.020231	0.0198930	0.0192790	0.01895300
1.5	0.021252	0.0230670	0.0230830	0.023314	0.0231580	0.0230880	0.02320900
2.0	0.021330	0.0232460	0.0232770	0.023291	0.0232120	0.0233110	0.02335400

Table 5: Coefficient Noise Standard Deviation (absolute value)

Standard Deviation of the mean of the solution $\mathbf{u}(N,\alpha)$. $\sigma = 1$							
$\alpha : N$	10	250	500	1000	10000	20000	50000
0.0	0.016823	0.012914	0.012201	0.010703	0.0061751	0.005965	0.0043796
0.5	0.016834	0.017617	0.016880	0.016391	0.0148880	0.013914	0.0131150
1.0	0.015286	0.017061	0.017248	0.017187	0.0175860	0.017433	0.0174830
1.5	0.013341	0.014550	0.014539	0.014610	0.0146610	0.014629	0.0146740
2.0	0.011676	0.012789	0.012741	0.012764	0.0129310	0.012764	0.0127850

becoming singular, while preserving the total energy of the noise. We present the results from the simulations for both the mean and standard variation on Table 4 and Table 5. In this case, due to the non-linearity of the problem, we observe dramatic difference in the mean of the solution for different values of α. When $\alpha = 0$ and $\alpha = 0.5$, the mean of the solution seems to converge to zero, however, for $\alpha \geq 1$ the solution seems to be approximately mesh independent. Similar behaviour of convergence vs. independence can be observed for the standard deviation. We can conclude, that for this non-linear example, the different choice of α results in a solution $u_\alpha(x)$ with completely different properties.

Another way to add noise to the coefficient of the Laplace equations is via an exponential function

$$\left(1 + e^{\eta(x,\alpha)}\right) u_{xx}(x) = x, \qquad u(0) = 0, \qquad u(1) = \frac{1}{6}. \tag{5}$$

Then we do the same numerical experiments. In this case, the computed solution has the same mean for all N and α (see Table 6), however, we again observe a difference in the variance of the total solution as well the individual Fourier coefficients. The results are given in Tables 7 and 8.

In all the previous examples we have kept the parameter $\sigma = 1$ constant. We can select σ as a function of α and force the noise $\eta(\sigma(\alpha), \alpha)$ to have uniform

Table 6: Coefficient Noise Mean (exponential)

Mean of the solution $\mathbf{u}(N,\alpha)$. $\sigma = 1$							
$\alpha : N$	10	250	500	1000	10000	20000	50000
0.0	0.018694	0.020815	0.020573	0.020793	0.020609	0.021526	0.020840
0.5	0.018892	0.020889	0.020770	0.021104	0.020502	0.020876	0.021065
1.0	0.018737	0.021071	0.020633	0.020854	0.021171	0.021109	0.020698
1.5	0.018719	0.020671	0.020726	0.021067	0.020876	0.020907	0.020951
2.0	0.019048	0.020853	0.020636	0.021118	0.021055	0.020900	0.020937

Table 7: Coefficient Noise Standard Deviation (exponential)

Standard Deviation of the mean of the solution $\mathbf{u}(N,\alpha)$. $\sigma = 1$							
$\alpha : N$	10	250	500	1000	10000	20000	50000
0.0	0.030039	0.039493	0.040165	0.040671	0.041306	0.041433	0.041522
0.5	0.023397	0.033686	0.034719	0.035781	0.038358	0.038961	0.039535
1.0	0.016694	0.021828	0.022596	0.022971	0.024646	0.024954	0.025189
1.5	0.012837	0.014632	0.014739	0.014891	0.014810	0.014957	0.014780
2.0	0.010849	0.012404	0.012266	0.012388	0.012352	0.012383	0.012344

variances for the mean $\sigma(\bar{\eta}(\sigma(\alpha),\alpha)) = 1$ for all values of α. To do that, we need to divide σ by the corresponding value of Table 1. Afterwards, we use the new "normalized" noise as input to (5). From the results give on Tables 9 and 10, we can see that even when we adjust the parameter σ, we still obtain solutions with significantly different statistical properties.

In another example we consider the steady state Burgers equation

$$\mu u_{xx} - u u_x = 2\mu - 2x^3 + \eta(\alpha), \qquad u(0) = 0, \qquad u(1) = 1, \tag{6}$$

where $\eta(\alpha)$ is $\frac{1}{f^\alpha}$ noise. The exact noise-free solution is $u(x) = x^2$. We take $\mu = 0.1$ to increase the influence of the non-linear term uu_x and we adjust $\sigma = 0.1$. Experimentally, we discovered that for larger σ, we would often generate noise that will make (6) ill conditioned and the PDE solver would fail to converge to a solution. Once again we observe very different results for both the standard deviation of the mean of the computed solution and the distribution of the corresponding Fourier frequencies.

6 Conclusion

While generating colored $\frac{1}{f^\alpha}$ noise is more expensive than simple white noise ($\alpha = 0$) we see that when applied to different problems, it can result solutions

Table 8: Coefficient noise Computed Coefficients (exponential)

Standard Deviation of the absolute value of individual Fourier coefficients $\mathbf{u}(N, \alpha)$. $\sigma = 1$						
$\alpha : k$	1	2	3	7	15	25
0.0	0.020852	0.0093093	0.0057344	0.00211260	0.00092783	0.00054521
0.5	0.021167	0.0089848	0.0055042	0.00202000	0.00088573	0.00052011
1.0	0.018968	0.0067651	0.0039406	0.00139180	0.00060170	0.00035156
1.5	0.013754	0.0052845	0.0029439	0.00101210	0.00043517	0.00025398
2.0	0.012037	0.0048039	0.0026727	0.00091579	0.00039355	0.00022966

Table 9: Coefficient Noise Standard Deviation (exponential, normalized)

Standard Deviation of the mean of the solution $\mathbf{u}(N, \alpha)$. Normalized σ.							
$\alpha : N$	10	250	500	1000	10000	20000	50000
0.0	0.030075	0.039482	0.040174	0.040519	0.041319	0.041439	0.041509
0.5	0.024348	0.034147	0.035700	0.036520	0.038858	0.039202	0.039893
1.0	0.019082	0.024499	0.025135	0.025709	0.026972	0.027444	0.027896
1.5	0.015951	0.018241	0.018451	0.018433	0.018444	0.018524	0.018489
2.0	0.015240	0.017119	0.017083	0.017207	0.017128	0.017225	0.017088

Table 10: Coefficient noise Computed Coefficients (exponential, normalized)

Standard Deviation of the absolute value of individual Fourier coefficients $\mathbf{u}(N, \alpha)$. Normalized σ.						
$\alpha : k$	1	2	3	7	15	25
0.0	0.020885	0.0093097	0.0057336	0.0021121	0.00092758	0.00054506
0.5	0.021234	0.0090597	0.0055517	0.0020381	0.00089377	0.00052487
1.0	0.019847	0.0073978	0.0043265	0.0015375	0.00066599	0.00038935
1.5	0.016138	0.0065030	0.0036361	0.0012553	0.00054011	0.00031526
2.0	0.015621	0.0067478	0.0037445	0.0012849	0.00055230	0.00032232

Table 11: Burgers Noise Standard Deviation

Standard Deviation of the mean of the solution $\mathbf{u}(N,\alpha)$. $\sigma = 0.1$							
$\alpha : N$	50	250	500	1000	10000	20000	50000
0.0	0.058221	0.057232	0.057255	0.057214	0.056915	0.057207	0.057698
0.5	0.051579	0.050788	0.050580	0.050707	0.050992	0.050192	0.050302
1.0	0.044053	0.042991	0.042799	0.043647	0.043269	0.042761	0.042914
1.5	0.036970	0.036124	0.036351	0.036163	0.035946	0.036111	0.035990
2.0	0.029671	0.028895	0.028673	0.028770	0.029347	0.028970	0.028771

Table 12: Computed Frequency Coefficients (Burgers Equations)

Standard Deviation of the absolute value of individual Fourier coefficients $\mathbf{u}(N,\alpha)$. $\mu = 0.1$, $\sigma = 0.1$						
$\alpha : k$	1	2	3	7	15	25
0.0	0.057698	0.0150620	0.0043334	$5.0102e-04$	$9.1792e-05$	$3.1157e-05$
0.5	0.050302	0.0099157	0.0024933	$2.1851e-04$	$3.0771e-05$	$9.0073e-06$
1.0	0.042914	0.0066314	0.0016356	$1.1768e-04$	$1.2587e-05$	$3.0027e-06$
1.5	0.035990	0.0045863	0.0012406	$8.7105e-05$	$7.9024e-06$	$1.6450e-06$
2.0	0.028771	0.0033501	0.0010229	$7.2993e-05$	$6.4266e-06$	$1.3120e-06$

with drastically different properties. It may be surprising to see that the choice of a particular kind of noise function can have such an influence on the solution. Given the number of examples of natural phenomena which seem to exhibit colored noise, and the fact that white noise has some unrealistic features, particularly as the sampling interval is decreased, it is reasonable to consider the use of colored noise when constructing and analyzing stochastic models.

Each of the examples presented here has considered a one-dimensional spatial domain for which a noise function was generated by **ff_alpha**, to be sampled at N equally spaced points, with standard deviation σ and parameter α. It is natural to consider the extension of this approach to cases in which the spatial domain is multidimensional, and in which there is time evolution as well.

The simplest observation is that all the remarks about a 1D spatial interval carry over unchanged to a system defined over a fixed 1D time interval.

If the domain is a product region of time and space, then standard Fourier transform methods allow us to construct a multidimensional noise function as the product of one dimensional noise functions; the component noise functions, in turn, are each defined by a number of sample points N_i, a standard deviation σ_i, and a value of α_i, which are then input to a multiple-FFT version of the algorithm **ff_alpha()**.

As far as the FFT computations are concerned, no distinction need be made between time and space dimensions. On the other hand, there may be good reasons to use different considerations for the value of σ_t and α_t associated with the time-wise noise component.

Some guidance in choosing the components of the noise parameters can be gained from considering the 1D case. Suppose, for instance, that a 3D spatial domain is being considered. Then the instantaneous energy of the noise signal, with parameters α_x, α_y and α_z, will be the same as that for a noise signal over a 1D region with parameter $\alpha = \sqrt{\alpha_x^2 + \alpha_y^2 + \alpha_z^2}$.

In the common case where there is no directional preference for the noise in the spatial dimensions, it is natural to choose common values of σ_x and α_x for all spatial noise components.

References

[1] Martin Gardner, *White and brown music, fractal curves and one-over-f fluctuations*, Scientific American, Volume 238, Number 4, April 1978, pages 16-32.

[2] Jeremy Kasdin, *Discrete Simulation of Colored Noise and Stochastic Processes and 1/f^a Power Law Noise Generation*, Proceedings of the IEEE, Volume 83, Number 5, 1995, pages 802-827.

[3] Edoardo Milotti, *1/f noise: a pedagogical review*, arXiv:physics/0204033.

[4] Sophocles Orfanidis, INTRODUCTION TO SIGNAL PROCESSING, Prentice-Hall, 1995.

[5] William Press, *Flicker Noises in Astronomy and Elsewhere*, Comments on Astrophysics, Volume 7, Number 4, 1978, pages 103-119.